Conventions, Accuracy Metrics, Classification, Regression

Nipun Batra and teaching staff

IIT Gandhinagar

August 7, 2025

Outline

Demo

Comet browser and automation of tasks

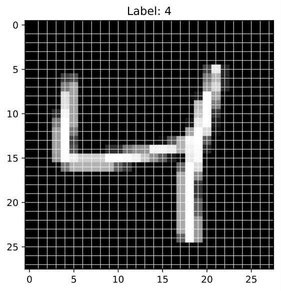
Revision: What is Machine Learning

"Field of study that gives computers the ability to learn without being explicitly programmed" - Arthur Samuel [1959]

Revision: What is Machine Learning

"Field of study that gives computers the ability to learn without being explicitly programmed" - Arthur Samuel [1959]
Let us work on the digit recognition problem.

Notebook: rule-based-vs-ml.html



Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.

Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.

Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.

Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.
- Maybe 4 can be thought of as: $| + — + | +$ another vertically down $|$

Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.
- Maybe 4 can be thought of as: $|$ + — + $|$ + another vertically down $|$

Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.
- Maybe 4 can be thought of as: $| + — + | +$ another vertically down $|$
- The heights of each of the $|$ need to be similar within tolerance

Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.
- Maybe 4 can be thought of as: $| + — + | +$ another vertically down $|$
- The heights of each of the $|$ need to be similar within tolerance

Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.
- Maybe 4 can be thought of as: $| + — + | +$ another vertically down $|$
- The heights of each of the $|$ need to be similar within tolerance
- Each of the $|$ can be slightly slanted. Similarly the horizontal line can be slanted.

Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.
- Maybe 4 can be thought of as: $| + — + | +$ another vertically down $|$
- The heights of each of the $|$ need to be similar within tolerance
- Each of the $|$ can be slightly slanted. Similarly the horizontal line can be slanted.

Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.
- Maybe 4 can be thought of as: $| + — + | +$ another vertically down $|$
- The heights of each of the $|$ need to be similar within tolerance
- Each of the $|$ can be slightly slanted. Similarly the horizontal line can be slanted.
- There can be some cases of 4 where the first $|$ is at 45 degrees

Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.
- Maybe 4 can be thought of as: $| + — + | +$ another vertically down $|$
- The heights of each of the $|$ need to be similar within tolerance
- Each of the $|$ can be slightly slanted. Similarly the horizontal line can be slanted.
- There can be some cases of 4 where the first $|$ is at 45 degrees

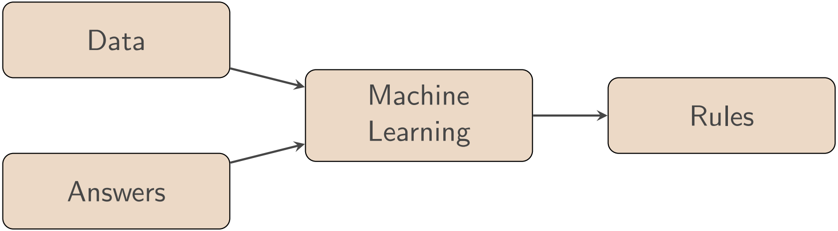
Revision: What is Machine Learning

- How would you program to recognise digits? Start with 4.
- Maybe 4 can be thought of as: $| + — + | +$ another vertically down $|$
- The heights of each of the $|$ need to be similar within tolerance
- Each of the $|$ can be slightly slanted. Similarly the horizontal line can be slanted.
- There can be some cases of 4 where the first $|$ is at 45 degrees
- There can be some cases of 4 where the width of each stroke is different

Traditional Programming vs Machine Learning

Traditional Programming

Machine Learning



Revision: What is Machine Learning

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." - Tom Mitchell

First ML Task: Grocery Store Tomato Quality Prediction

Problem statement: You want to predict the quality of a tomato given its visual features.

Dataset

Imagine you have some past data on quality of tomatoes. What visual features do you think will be useful?

Dataset

Imagine you have some past data on quality of tomatoes. What visual features do you think will be useful?

Dataset

Imagine you have some past data on quality of tomatoes. What visual features do you think will be useful?

- Size

Dataset

Imagine you have some past data on quality of tomatoes. What visual features do you think will be useful?

- Size

Dataset

Imagine you have some past data on quality of tomatoes. What visual features do you think will be useful?

- Size
- Colour

Dataset

Imagine you have some past data on quality of tomatoes. What visual features do you think will be useful?

- Size
- Colour

Dataset

Imagine you have some past data on quality of tomatoes. What visual features do you think will be useful?

- Size
- Colour
- Texture

Sample Dataset

Here is our example dataset with tomato features:

Sample	**Colour**	**Size**	**Texture**	**Condition**
1	Orange	Small	Smooth	Good
2	Red	Small	Rough	Good
3	Orange	Medium	Smooth	Bad
4	Yellow	Large	Smooth	Bad

Pop Quiz #1

Quick Quiz 1

Is the sample number a useful feature for predicting quality of a tomato?

Pop Quiz #1

Quick Quiz 1

Is the sample number a useful feature for predicting quality of a tomato?
Answer: Usually no! Sample numbers are typically arbitrary identifiers and not meaningful features. Let us remove it.

Pop Quiz #1

Quick Quiz 1

When could sample number be useful?

Pop Quiz #1

Quick Quiz 1

When could sample number be useful?
In some cases, the sample number might be useful for tracking or auditing purposes. E.g. if some trucks are delayed during delivery, the sample number could help identify which batch of tomatoes was affected.

Useful Features

Useful Features

Let us modify our data table for now.

Colour	**Size**	**Texture**	**Condition**
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad

Training Set

Colour	**Size**	**Texture**	**Condition**
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad

Training Set

Colour	**Size**	**Texture**	**Condition**
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad

The training set consists of two parts:

Training Set

Colour	Size	Texture	Condition
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad

The training set consists of two parts:

Training Set

Colour	**Size**	**Texture**	**Condition**
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad

The training set consists of two parts:

1. Features (Input Variables)

Training Set

Colour	**Size**	**Texture**	**Condition**
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad

The training set consists of two parts:

1. Features (Input Variables)

Training Set

Colour	**Size**	**Texture**	**Condition**
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad

The training set consists of two parts:

1. Features (Input Variables)
2. Output or Response Variable

Training Set

Colour	**Size**	**Texture**	**Condition**
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad

Training Set

Colour	**Size**	**Texture**	**Condition**
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad

Computers work with numbers! We need to encode categorical data numerically (one-hot encoding):

C0	**C1**	**S0**	**S1**	**T0**	**T1**	**Good?**
0	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	0	1	1	0	0
1	0	0	0	1	0	0

Training Set

Colour	Size	Texture	Condition
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad

Computers work with numbers! We need to encode categorical data numerically (one-hot encoding):

C0	C1	S0	S1	T0	T1	Good?
0	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	0	1	1	0	0
1	0	0	0	1	0	0

Orange=00, Red=01, Yellow=10; Small=10, Medium=01, Large=00; Smooth=10, Rough=01; Good=1, Bad=0

More details on encoding later!

Data Matrix

C0	C1	S0	S1	T0	T1	Good?
0	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	0	1	1	0	0
1	0	0	0	1	0	0

Data Matrix

C0	C1	S0	S1	T0	T1	Good?
0	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	0	1	1	0	0
1	0	0	0	1	0	0

We call this data matrix $\mathbf{X}$, and the complete dataset $\mathcal{D}$:

1. Feature matrix ($\mathbf{X} \in \mathbb{R}^{n \times d}$) containing data of n samples each of which is d dimensional.

Data Matrix

C0	C1	S0	S1	T0	T1	Good?
0	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	0	1	1	0	0
1	0	0	0	1	0	0

We call this data matrix $\mathbf{X}$, and the complete dataset $\mathcal{D}$:

1. Feature matrix ($\mathbf{X} \in \mathbb{R}^{n \times d}$) containing data of n samples each of which is d dimensional.
2. Output vector ($\mathbf{y} \in \mathbb{R}^{n}$) containing output variable for n samples.

Data Matrix

C0	C1	S0	S1	T0	T1	Good?
0	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	0	1	1	0	0
1	0	0	0	1	0	0

We call this data matrix $\mathbf{X}$, and the complete dataset $\mathcal{D}$:

1. Feature matrix ($\mathbf{X} \in \mathbb{R}^{n \times d}$) containing data of n samples each of which is d dimensional.
2. Output vector ($\mathbf{y} \in \mathbb{R}^{n}$) containing output variable for n samples.
3. Complete dataset: $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ (set of sample-label pairs)

Data Matrix

C0	C1	S0	S1	T0	T1	Good?
0	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	0	1	1	0	0
1	0	0	0	1	0	0

We call this data matrix $\mathbf{X}$, and the complete dataset $\mathcal{D}$:

1. Feature matrix ($\mathbf{X} \in \mathbb{R}^{n \times d}$) containing data of n samples each of which is d dimensional.
2. Output vector ($\mathbf{y} \in \mathbb{R}^{n}$) containing output variable for n samples.
3. Complete dataset: $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ (set of sample-label pairs)

Data Matrix

C0	C1	S0	S1	T0	T1	Good?
0	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	0	1	1	0	0
1	0	0	0	1	0	0

We call this data matrix $\mathbf{X}$, and the complete dataset $\mathcal{D}$:

1. Feature matrix ($\mathbf{X} \in \mathbb{R}^{n \times d}$) containing data of n samples each of which is d dimensional.
2. Output vector ($\mathbf{y} \in \mathbb{R}^{n}$) containing output variable for n samples.
3. Complete dataset: $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ (set of sample-label pairs)

For this example: $n = 4$ (samples), $d = 6$ (features after one-hot encoding)

Mathematical Notation Convention

Important: Mathematical Notation Convention

Matrices use **bold uppercase** ($\mathbf{X}$), vectors use **bold lowercase** ($\mathbf{y}$), scalars use regular letters (n, d)

Example: Examples from Our Tomato Dataset

- **Scalars**: $n = 4$ (samples), $d = 6$ (features), $y_1 = 1$

Convention: We write $[a, b, c]^\top$ instead of $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ to save space

Mathematical Notation Convention

Important: Mathematical Notation Convention

Matrices use **bold uppercase** ($\mathbf{X}$), vectors use **bold lowercase** ($\mathbf{y}$), scalars use regular letters (n, d)

Example: Examples from Our Tomato Dataset

- **Scalars**: $n = 4$ (samples), $d = 6$ (features), $y_1 = 1$

Convention: We write $[a, b, c]^\top$ instead of $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ to save space

Mathematical Notation Convention

Important: Mathematical Notation Convention

Matrices use **bold uppercase** ($\mathbf{X}$), vectors use **bold lowercase** ($\mathbf{y}$), scalars use regular letters (n, d)

Example: Examples from Our Tomato Dataset

- **Scalars**: $n = 4$ (samples), $d = 6$ (features), $y_1 = 1$

Convention: We write $[a, b, c]^\top$ instead of $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ to save space

Mathematical Notation Convention

Important: Mathematical Notation Convention

Matrices use **bold uppercase** ($\mathbf{X}$), vectors use **bold lowercase** ($\mathbf{y}$), scalars use regular letters (n, d)

Example: Examples from Our Tomato Dataset

- **Scalars**: $n = 4$ (samples), $d = 6$ (features), $y_1 = 1$
- **Vectors**: $\mathbf{y} = [1, 1, 0, 0]^\top$, $\mathbf{x}_1 = [0, 0, 1, 0, 1, 0]^\top$

Convention: We write $[a, b, c]^\top$ instead of $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ to save space

Mathematical Notation Convention

Important: Mathematical Notation Convention

Matrices use **bold uppercase** ($\mathbf{X}$), vectors use **bold lowercase** ($\mathbf{y}$), scalars use regular letters (n, d)

Example: Examples from Our Tomato Dataset

- **Scalars**: $n = 4$ (samples), $d = 6$ (features), $y_1 = 1$
- **Vectors**: $\mathbf{y} = [1, 1, 0, 0]^\top$, $\mathbf{x}_1 = [0, 0, 1, 0, 1, 0]^\top$

Convention: We write $[a, b, c]^\top$ instead of $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ to save space

Mathematical Notation Convention

Important: Mathematical Notation Convention

Matrices use **bold uppercase** ($\mathbf{X}$), vectors use **bold lowercase** ($\mathbf{y}$), scalars use regular letters (n, d)

Example: Examples from Our Tomato Dataset

- **Scalars**: $n = 4$ (samples), $d = 6$ (features), $y_1 = 1$
- **Vectors**: $\mathbf{y} = [1, 1, 0, 0]^\top$, $\mathbf{x}_1 = [0, 0, 1, 0, 1, 0]^\top$
- **Matrices**: $\mathbf{X} \in \mathbb{R}^{4\times 6}$ (feature matrix)

Mathematical Notation Convention

Important: Mathematical Notation Convention

Matrices use **bold uppercase** ($\mathbf{X}$), vectors use **bold lowercase** ($\mathbf{y}$), scalars use regular letters (n, d)

Example: Examples from Our Tomato Dataset

- **Scalars**: $n = 4$ (samples), $d = 6$ (features), $y_1 = 1$
- **Vectors**: $\mathbf{y} = [1, 1, 0, 0]^\top$, $\mathbf{x}_1 = [0, 0, 1, 0, 1, 0]^\top$
- **Matrices**: $\mathbf{X} \in \mathbb{R}^{4\times 6}$ (feature matrix)

Convention: We write $[a, b, c]^\top$ instead of $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ to save space

Data Matrix Details

- Feature matrix: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}$ where $\mathbf{x}_i \in \mathbb{R}^d$

Data Matrix Details

- Feature matrix: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}$ where $\mathbf{x}_i \in \mathbb{R}^d$

Data Matrix Details

- Feature matrix: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}$ where $\mathbf{x}_i \in \mathbb{R}^d$

Data Matrix Details

- Feature matrix: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}$ where $\mathbf{x}_i \in \mathbb{R}^d$
- Example: $\mathbf{x}_1 = [0, 0, 1, 0, 1, 0]^\top$ (Orange=00, Small=10, Smooth=10)

Data Matrix Details

- Feature matrix: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}$ where $\mathbf{x}_i \in \mathbb{R}^d$
- Example: $\mathbf{x}_1 = [0, 0, 1, 0, 1, 0]^\top$ (Orange=00, Small=10, Smooth=10)

Data Matrix Details

- Feature matrix: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}$ where $\mathbf{x}_i \in \mathbb{R}^d$
- Example: $\mathbf{x}_1 = [0, 0, 1, 0, 1, 0]^\top$ (Orange=00, Small=10, Smooth=10)
- Complete dataset: $\mathcal{D} = \{(\mathbf{x}_i^\top, y_i)\}_{i=1}^n$

Data Matrix Details

- Feature matrix: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}$ where $\mathbf{x}_i \in \mathbb{R}^d$
- Example: $\mathbf{x}_1 = [0, 0, 1, 0, 1, 0]^\top$ (Orange=00, Small=10, Smooth=10)
- Complete dataset: $\mathcal{D} = \{(\mathbf{x}_i^\top, y_i)\}_{i=1}^n$

For this example: $n = 4$, $d = 6$, so $\mathbf{X} \in \mathbb{R}^{4\times 6}$ and $\mathbf{y} \in \mathbb{R}^4$

Prediction Task

Estimate condition for unseen tomatoes (#5, 6) based on data set.

Colour	**Size**	**Texture**	**Condition**
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad
Red	Large	Rough	?
Orange	Large	Rough	?

Testing Set

Testing set is similar to training set, but, does not contain labels for output variable.

Colour	Size	Texture	Condition
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad
Red	Large	Rough	?
Orange	Large	Rough	?

Prediction Task

We hope to:

Prediction Task

We hope to:

Prediction Task

We hope to:

1. Learn function f: $y = f(\mathbf{x})$ where $\mathbf{x}$ represents features

Prediction Task

We hope to:

1. Learn function f: $y = f(\mathbf{x})$ where $\mathbf{x}$ represents features

Prediction Task

We hope to:

1. Learn function f: $y = f(\mathbf{x})$ where $\mathbf{x}$ represents features
2. From training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$

Prediction Task

We hope to:

1. Learn function f: $y = f(\mathbf{x})$ where $\mathbf{x}$ represents features
2. From training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$

Prediction Task

We hope to:

1. Learn function f: $y = f(\mathbf{x})$ where $\mathbf{x}$ represents features
2. From training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$
3. Predict condition for new test samples

Prediction Task

We hope to:

1. Learn function f: $y = f(\mathbf{x})$ where $\mathbf{x}$ represents features
2. From training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$
3. Predict condition for new test samples

Colour	**Size**	**Texture**	**Condition**
Orange	Small	Smooth	Good
Red	Small	Rough	Good
Orange	Medium	Smooth	Bad
Yellow	Large	Smooth	Bad
Red	Large	Rough	?
Orange	Large	Rough	?

Training set used to learn f, **Test set** for predictions

Generalisation

Important: Key Question

Is predicting well on our test set enough to say our model generalises?

Generalisation

Important: Key Question

Is predicting well on our test set enough to say our model generalises?

- **Answer**: Ideally, no!

Generalisation

Important: Key Question

Is predicting well on our test set enough to say our model generalises?

- **Answer**: Ideally, no!

Generalisation

Important: Key Question

Is predicting well on our test set enough to say our model generalises?

- **Answer**: Ideally, no!

Generalisation

Important: Key Question

Is predicting well on our test set enough to say our model generalises?

- **Answer**: Ideally, no!
- **Goal**: We want to predict well on *all possible future inputs*

Generalisation

Important: Key Question

Is predicting well on our test set enough to say our model generalises?

- **Answer**: Ideally, no!
- **Goal**: We want to predict well on *all possible future inputs*

Generalisation

Important: Key Question

Is predicting well on our test set enough to say our model generalises?

- **Answer**: Ideally, no!
- **Goal**: We want to predict well on *all possible future inputs*
- **Reality**: Test set is only a *sample* from all possible inputs

Generalisation

Important: Key Question

Is predicting well on our test set enough to say our model generalises?

- **Answer**: Ideally, no!
- **Goal**: We want to predict well on *all possible future inputs*
- **Reality**: Test set is only a *sample* from all possible inputs

Generalisation

Important: Key Question

Is predicting well on our test set enough to say our model generalises?

- **Answer**: Ideally, no!
- **Goal**: We want to predict well on *all possible future inputs*
- **Reality**: Test set is only a *sample* from all possible inputs
- **Assumption**: Test set is *representative* of the true data distribution

Generalisation

Important: Key Question

Is predicting well on our test set enough to say our model generalises?

- **Answer**: Ideally, no!
- **Goal**: We want to predict well on *all possible future inputs*
- **Reality**: Test set is only a *sample* from all possible inputs
- **Assumption**: Test set is *representative* of the true data distribution

Key Points

Generalisation = Performance on **unseen data** from the same distribution as training data

Sample vs Population: A Simple Example

Example: Tomato Farm: 10,000 tomatoes ready for harvest

- **Population**: All 10,000 tomatoes

Key Challenge: Will your 100 tomatoes represent all 10,000? What if you only picked from one corner?

ML Connection: Training/test sets are samples from a larger population of all possible data

Sample vs Population: A Simple Example

Example: Tomato Farm: 10,000 tomatoes ready for harvest

- **Population**: All 10,000 tomatoes

Key Challenge: Will your 100 tomatoes represent all 10,000? What if you only picked from one corner?

ML Connection: Training/test sets are samples from a larger population of all possible data

Sample vs Population: A Simple Example

Example: Tomato Farm: 10,000 tomatoes ready for harvest

- **Population**: All 10,000 tomatoes

Key Challenge: Will your 100 tomatoes represent all 10,000? What if you only picked from one corner?

ML Connection: Training/test sets are samples from a larger population of all possible data

Sample vs Population: A Simple Example

Example: Tomato Farm: 10,000 tomatoes ready for harvest

- **Population**: All 10,000 tomatoes
- **Sample**: You inspect 100 random tomatoes

Key Challenge: Will your 100 tomatoes represent all 10,000? What if you only picked from one corner?

ML Connection: Training/test sets are samples from a larger population of all possible data

Sample vs Population: A Simple Example

Example: Tomato Farm: 10,000 tomatoes ready for harvest

- **Population**: All 10,000 tomatoes
- **Sample**: You inspect 100 random tomatoes

Key Challenge: Will your 100 tomatoes represent all 10,000? What if you only picked from one corner?

ML Connection: Training/test sets are samples from a larger population of all possible data

Sample vs Population: A Simple Example

Example: Tomato Farm: 10,000 tomatoes ready for harvest

- **Population**: All 10,000 tomatoes
- **Sample**: You inspect 100 random tomatoes
- **Goal**: Make decisions about all 10,000 based on your 100

Sample vs Population: A Simple Example

Example: Tomato Farm: 10,000 tomatoes ready for harvest

- **Population**: All 10,000 tomatoes
- **Sample**: You inspect 100 random tomatoes
- **Goal**: Make decisions about all 10,000 based on your 100

Key Challenge: Will your 100 tomatoes represent all 10,000? What if you only picked from one corner?

Sample vs Population: A Simple Example

Example: Tomato Farm: 10,000 tomatoes ready for harvest

- **Population**: All 10,000 tomatoes
- **Sample**: You inspect 100 random tomatoes
- **Goal**: Make decisions about all 10,000 based on your 100

Key Challenge: Will your 100 tomatoes represent all 10,000? What if you only picked from one corner?
ML Connection: Training/test sets are samples from a larger population of all possible data

From Tomatoes to Machine Learning

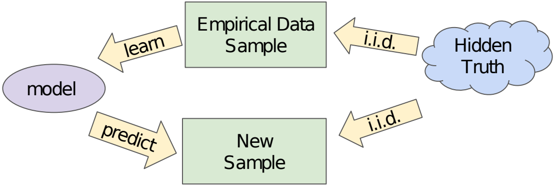


Image courtesy Google ML crash course

From Tomatoes to Machine Learning

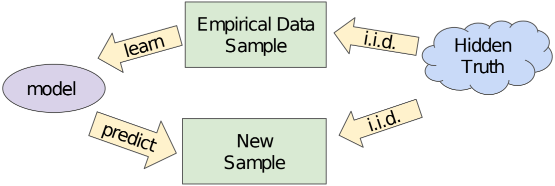


Image courtesy Google ML crash course

The ML Connection:

- **Population**: All possible tomato data (past, present, future)

From Tomatoes to Machine Learning

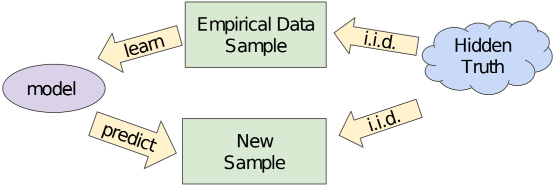


Image courtesy Google ML crash course

The ML Connection:

- **Population**: All possible tomato data (past, present, future)

From Tomatoes to Machine Learning

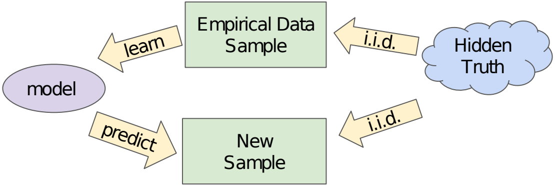


Image courtesy Google ML crash course

The ML Connection:

- **Population**: All possible tomato data (past, present, future)

From Tomatoes to Machine Learning

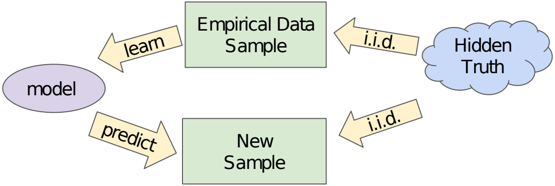


Image courtesy Google ML crash course

The ML Connection:

- **Population**: All possible tomato data (past, present, future)
- **Training set**: Our 4 tomato samples (like picking from one area)

From Tomatoes to Machine Learning

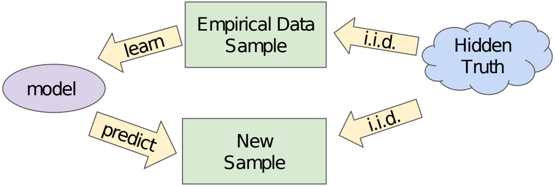


Image courtesy Google ML crash course

The ML Connection:

- **Population**: All possible tomato data (past, present, future)
- **Training set**: Our 4 tomato samples (like picking from one area)

From Tomatoes to Machine Learning

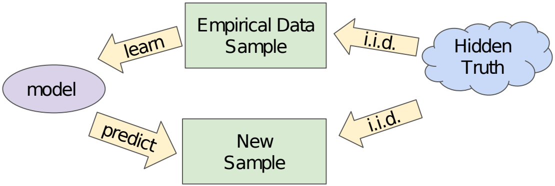


Image courtesy Google ML crash course

The ML Connection:

- **Population**: All possible tomato data (past, present, future)
- **Training set**: Our 4 tomato samples (like picking from one area)
- **Test set**: 2 new samples (like picking from another area)

From Tomatoes to Machine Learning

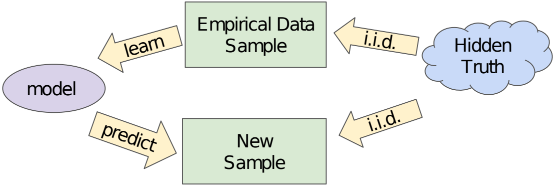


Image courtesy Google ML crash course

The ML Connection:

- **Population**: All possible tomato data (past, present, future)
- **Training set**: Our 4 tomato samples (like picking from one area)
- **Test set**: 2 new samples (like picking from another area)

Generalisation goal: Will our model work on *all future tomatoes*, not just our small samples?

Second ML Task: Campus Energy Prediction

Regression Problem: Predicting continuous energy consumption (kWh)
Key factors: # People, Temperature

# People	Temp (°C)	Energy (kWh)
4000	30	30
4200	30	32
4200	35	40
3000	20	?
1000	45	?

Second ML Task: Campus Energy Prediction

Regression Problem: Predicting continuous energy consumption (kWh)

Key factors: # People, Temperature

# People	Temp (°C)	Energy (kWh)
4000	30	30
4200	30	32
4200	35	40
3000	20	?
1000	45	?

Difference from tomatoes: Energy is *continuous*, not categories

Classification vs Regression

- Classification

Classification vs Regression

- Classification

Classification vs Regression

- Classification

Classification vs Regression

- Classification
  - Output variable is discrete

Classification vs Regression

- Classification
  - Output variable is discrete

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \dots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)
- Regression

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)
- Regression

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)
- Regression
  - Output variable is continuous

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)
- Regression
  - Output variable is continuous

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)
- Regression
  - Output variable is continuous
  - i.e. $y_i \in \mathbb{R}$

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)
- Regression
  - Output variable is continuous
  - i.e. $y_i \in \mathbb{R}$

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)
- Regression
  - Output variable is continuous
  - i.e. $y_i \in \mathbb{R}$
  - Examples - Predicting:

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)
- Regression
  - Output variable is continuous
  - i.e. $y_i \in \mathbb{R}$
  - Examples - Predicting:

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)
- Regression
  - Output variable is continuous
  - i.e. $y_i \in \mathbb{R}$
  - Examples - Predicting:
    - How much energy will campus consume?

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)
- Regression
  - Output variable is continuous
  - i.e. $y_i \in \mathbb{R}$
  - Examples - Predicting:
    - How much energy will campus consume?

Classification vs Regression

- Classification
  - Output variable is discrete
  - i.e. $y_i \in \{1, 2, \ldots, k\}$ where k is number of classes
  - Examples - Predicting:
    - Will I get a loan? (Yes, No)
    - What is the quality of fruit? (Good, Bad)
- Regression
  - Output variable is continuous
  - i.e. $y_i \in \mathbb{R}$
  - Examples - Predicting:
    - How much energy will campus consume?
    - How much rainfall will fall?

Pop Quiz #2

Quick Quiz 2

Which of these is a regression problem?

- a) Predicting if an email is spam or not

Pop Quiz #2

Quick Quiz 2

Which of these is a regression problem?

- a) Predicting if an email is spam or not
- b) Classifying images as cat, dog, or bird

Pop Quiz #2

Quick Quiz 2

Which of these is a regression problem?

- a) Predicting if an email is spam or not
- b) Classifying images as cat, dog, or bird
- c) Predicting house prices

Pop Quiz #2

Quick Quiz 2

Which of these is a regression problem?

- a) Predicting if an email is spam or not
- b) Classifying images as cat, dog, or bird
- c) Predicting house prices
- d) Determining if a tumor is malignant or benign

Pop Quiz #2 - Answer

Answer: c) House prices are continuous values - that's regression!

Now We Have Models - But How Good Are They?

- We've learned to predict tomato quality (classification)

Now We Have Models - But How Good Are They?

- We've learned to predict tomato quality (classification)
- We've learned to predict energy consumption (regression)

Now We Have Models - But How Good Are They?

- We've learned to predict tomato quality (classification)
- We've learned to predict energy consumption (regression)

Now We Have Models - But How Good Are They?

- We've learned to predict tomato quality (classification)
- We've learned to predict energy consumption (regression)

Now We Have Models - But How Good Are They?

- We've learned to predict tomato quality (classification)
- We've learned to predict energy consumption (regression)
- **Key question**: How do we measure if our predictions are good?

Now We Have Models - But How Good Are They?

- We've learned to predict tomato quality (classification)
- We've learned to predict energy consumption (regression)
- **Key question**: How do we measure if our predictions are good?

Example: The Challenge

If our model predicts 5 tomatoes correctly and 3 incorrectly, is that good or bad? We need metrics!

Now We Have Models - But How Good Are They?

- We've learned to predict tomato quality (classification)
- We've learned to predict energy consumption (regression)
- **Key question**: How do we measure if our predictions are good?

Example: The Challenge

If our model predicts 5 tomatoes correctly and 3 incorrectly, is that good or bad? We need metrics!

Coming up: Different metrics for classification vs regression problems

Back to Our Tomato Example: How Did We Do?

Let's say we trained our model and tested it on 5 new tomatoes:

#	Actual	Predicted
1	Good	Good
2	Good	Good
3	Bad	Good
4	Bad	Good
5	Bad	Bad

Back to Our Tomato Example: How Did We Do?

Let's say we trained our model and tested it on 5 new tomatoes:

#	Actual	Predicted
1	Good	Good
2	Good	Good
3	Bad	Good
4	Bad	Good
5	Bad	Bad

Questions:

- How many did we get right? How many wrong?

Back to Our Tomato Example: How Did We Do?

Let's say we trained our model and tested it on 5 new tomatoes:

#	Actual	Predicted
1	Good	Good
2	Good	Good
3	Bad	Good
4	Bad	Good
5	Bad	Bad

Questions:

- How many did we get right? How many wrong?
- Is 3 out of 5 correct "good enough"?

Back to Our Tomato Example: How Did We Do?

Let's say we trained our model and tested it on 5 new tomatoes:

#	Actual	Predicted
1	Good	Good
2	Good	Good
3	Bad	Good
4	Bad	Good
5	Bad	Bad

Questions:

- How many did we get right? How many wrong?
- Is 3 out of 5 correct "good enough"?
- What if getting a bad tomato wrong is worse than getting a good tomato wrong?

Organizing Our Results

Let's organize the predictions in a simpler way:

$$\text{Model Predicted } (\hat{\mathbf{y}}) \quad \begin{pmatrix} \text{Good} \\ \text{Good} \\ \text{Good} \\ \text{Good} \\ \text{Bad} \end{pmatrix} \qquad \text{Actually Was } (\mathbf{y}) \quad \begin{pmatrix} \text{Good} \\ \text{Good} \\ \text{Bad} \\ \text{Bad} \\ \text{Bad} \end{pmatrix}$$

Organizing Our Results

Let's organize the predictions in a simpler way:

$$\begin{array}{cc} \text{Model Predicted } (\hat{\mathbf{y}}) & \text{Actually Was } (\mathbf{y}) \\ \begin{pmatrix} \text{Good} \\ \text{Good} \\ \text{Good} \\ \text{Good} \\ \text{Bad} \end{pmatrix} & \begin{pmatrix} \text{Good} \\ \text{Good} \\ \text{Bad} \\ \text{Bad} \\ \text{Bad} \end{pmatrix} \end{array}$$

Each row = one tomato's result
Goal: Create systematic ways to measure performance from these comparisons

Converting to Numbers for Computation

Remember: Computers work with numbers! Let's encode our categories:

Example: Binary Encoding

Bad $= 0$, **Good** $= 1$

Now our results become:

$$\text{Predicted } (\hat{y}) \quad \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \qquad \text{Ground Truth } (\mathbf{y}) \quad \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Ground Truth $=$ The correct answers (what actually happened)

Accuracy: Measuring Overall Performance

How many predictions did we get exactly right?

$$\text{Predicted } (\hat{\mathbf{y}}) \quad \begin{array}{c} \checkmark \\ \checkmark \\ \\ \\ \checkmark \end{array} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \qquad \text{Ground Truth } (\mathbf{y}) \quad \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Accuracy: Measuring Overall Performance

How many predictions did we get exactly right?

$$
\begin{array}{c}
\text{Predicted } (\hat{\mathbf{y}}) \\
\begin{array}{c} \checkmark \\ \checkmark \\ \\ \\ \checkmark \end{array}
\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}
\end{array}
\qquad
\begin{array}{c}
\text{Ground Truth } (\mathbf{y}) \\
\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}
\end{array}
$$

Definition: Accuracy Formula

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}}$$

Accuracy: Measuring Overall Performance

How many predictions did we get exactly right?

$$\begin{array}{cc} & \text{Predicted } (\hat{\mathbf{y}}) \\ \begin{array}{c} \checkmark \\ \checkmark \\ \\ \\ \checkmark \end{array} & \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \end{array} \qquad \begin{array}{c} \text{Ground Truth } (\mathbf{y}) \\ \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array}$$

Definition: Accuracy Formula

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}}$$

For our example: Accuracy $= \frac{3}{5} = 0.6$ or 60%

Mathematical Notation: Two Ways to Count Correct Predictions

Mathematical Notation: Two Ways to Count Correct Predictions

Mathematical Notation: Two Ways to Count Correct Predictions

- **Set cardinality notation:** $|\{i : y_i = \hat{y}_i\}|$
  - Reads as: "Number of indices i such that $y_i = \hat{y}_i$"

Mathematical Notation: Two Ways to Count Correct Predictions

- **Set cardinality notation:** $|\{i : y_i = \hat{y}_i\}|$
  - Reads as: "Number of indices i such that $y_i = \hat{y}_i$"
  - Counts how many samples satisfy the condition

Mathematical Notation: Two Ways to Count Correct Predictions

- **Set cardinality notation:** $|\{i : y_i = \hat{y}_i\}|$
  - Reads as: "Number of indices i such that $y_i = \hat{y}_i$"
  - Counts how many samples satisfy the condition

Mathematical Notation: Two Ways to Count Correct Predictions

- **Set cardinality notation:** $|\{i : y_i = \hat{y}_i\}|$
  - Reads as: "Number of indices i such that $y_i = \hat{y}_i$"
  - Counts how many samples satisfy the condition

Mathematical Notation: Two Ways to Count Correct Predictions

- **Set cardinality notation:** $|\{i : y_i = \hat{y}_i\}|$
  - Reads as: "Number of indices i such that $y_i = \hat{y}_i$"
  - Counts how many samples satisfy the condition
- **Alternative: Indicator function notation**

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} \mathbf{1}[y_i = \hat{y}_i]}{n}$$

$$\text{where } \mathbf{1}[\text{condition}] = \begin{cases} 1 & \text{if condition is true} \\ 0 & \text{if condition is false} \end{cases}$$

Mathematical Notation: Two Ways to Count Correct Predictions

- **Set cardinality notation:** $|\{i : y_i = \hat{y}_i\}|$
  - Reads as: "Number of indices i such that $y_i = \hat{y}_i$"
  - Counts how many samples satisfy the condition
- **Alternative: Indicator function notation**

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} \mathbf{1}[y_i = \hat{y}_i]}{n}$$

$$\text{where } \mathbf{1}[\text{condition}] = \begin{cases} 1 & \text{if condition is true} \\ 0 & \text{if condition is false} \end{cases}$$

Mathematical Notation: Two Ways to Count Correct Predictions

- **Set cardinality notation:** $|\{i : y_i = \hat{y}_i\}|$
  - Reads as: "Number of indices i such that $y_i = \hat{y}_i$"
  - Counts how many samples satisfy the condition
- **Alternative: Indicator function notation**

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} \mathbf{1}[y_i = \hat{y}_i]}{n}$$

where $\mathbf{1}[\text{condition}] = \begin{cases} 1 & \text{if condition is true} \\ 0 & \text{if condition is false} \end{cases}$

- Both notations are mathematically equivalent and commonly used in ML literature

Two Views: Predictions vs Confusion Matrix

Two Views: Predictions vs Confusion Matrix

Model Predictions

#	Actual	Predicted
1	Good	Good
2	Good	Good
3	Bad	Good
4	Bad	Good
5	Bad	Bad

Two Views: Predictions vs Confusion Matrix

Model Predictions

#	Actual	Predicted
1	Good	Good
2	Good	Good
3	Bad	Good
4	Bad	Good
5	Bad	Bad

Confusion Matrix

		Bad	Good
Pred	**Bad**	1	0
	Good	2	2

Confusion Matrix: Understanding the Four Outcomes

Confusion Matrix		**Ground Truth**	
		Positive	**Negative**
Predicted	**Positive**	**TP**	**FP**
	Negative	**FN**	**TN**

Definition: Four Outcomes

- **TP (True Positive):** Correctly predicted positive

Confusion Matrix: Understanding the Four Outcomes

Confusion Matrix		Ground Truth	
		Positive	**Negative**
Predicted	**Positive**	**TP**	**FP**
	Negative	**FN**	**TN**

Definition: Four Outcomes

- **TP (True Positive):** Correctly predicted positive
- **TN (True Negative):** Correctly predicted negative

Confusion Matrix: Understanding the Four Outcomes

Confusion Matrix		Ground Truth	
		Positive	**Negative**
Predicted	**Positive**	**TP**	**FP**
	Negative	**FN**	**TN**

Definition: Four Outcomes

- **TP (True Positive):** Correctly predicted positive
- **TN (True Negative):** Correctly predicted negative
- **FP (False Positive):** Wrong! Said positive but actually negative

Confusion Matrix: Understanding the Four Outcomes

Confusion Matrix		Ground Truth	
		Positive	**Negative**
Predicted	**Positive**	**TP**	**FP**
	Negative	**FN**	**TN**

Definition: Four Outcomes

- **TP (True Positive):** Correctly predicted positive
- **TN (True Negative):** Correctly predicted negative
- **FP (False Positive):** Wrong! Said positive but actually negative
- **FN (False Negative):** Wrong! Said negative but actually positive

Confusion Matrix: Precision Focus

Confusion Matrix		Ground Truth		Row Totals
		Positive	**Negative**	
Predicted	**Positive**	TP	FP	**TP + FP**
	Negative	FN	TN	**FN + TN**
		TP + FN	**FP + TN**	**Total**

Example: Focus: Predicted Positives

$$\text{Precision} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FP}}$$

"Of all predicted positives, how many were actually positive?"

Confusion Matrix: Recall Focus

Confusion Matrix		Ground Truth		Row Totals
		Positive	**Negative**	
Predicted	**Positive**	TP	FP	**TP + FP**
	Negative	FN	TN	**FN + TN**
		TP + FN	**FP + TN**	**Total**

Example: Focus: Actual Positives

$$\text{Recall} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}}$$

"Of all actual positives, how many did I catch?"

When Classes Are Imbalanced

Many datasets have unequal class distributions!

Example: Example: Medical Screening

Out of 1000 patients tested:

- **990** patients are healthy (negative class)

When Classes Are Imbalanced

Many datasets have unequal class distributions!

Example: Example: Medical Screening

Out of 1000 patients tested:

- **990** patients are healthy (negative class)
- **10** patients have the disease (positive class)

When Classes Are Imbalanced

Many datasets have unequal class distributions!

Example: Example: Medical Screening

Out of 1000 patients tested:

- **990** patients are healthy (negative class)
- **10** patients have the disease (positive class)

When Classes Are Imbalanced

Many datasets have unequal class distributions!

Example: Example: Medical Screening

Out of 1000 patients tested:

- **990** patients are healthy (negative class)
- **10** patients have the disease (positive class)

Key Points

Why is this a problem?

- A "lazy" classifier that always predicts "healthy" gets 99% accuracy!

When Classes Are Imbalanced

Many datasets have unequal class distributions!

Example: Example: Medical Screening

Out of 1000 patients tested:

- **990** patients are healthy (negative class)
- **10** patients have the disease (positive class)

Key Points

Why is this a problem?

- A “lazy” classifier that always predicts “healthy” gets 99% accuracy!
- But it completely misses all disease cases

When Classes Are Imbalanced

Many datasets have unequal class distributions!

Example: Example: Medical Screening

Out of 1000 patients tested:

- **990** patients are healthy (negative class)
- **10** patients have the disease (positive class)

Key Points

Why is this a problem?

- A "lazy" classifier that always predicts "healthy" gets 99% accuracy!
- But it completely misses all disease cases
- Accuracy alone becomes misleading

When Classes Are Imbalanced

Many datasets have unequal class distributions!

Example: Example: Medical Screening

Out of 1000 patients tested:

- **990** patients are healthy (negative class)
- **10** patients have the disease (positive class)

Key Points

Why is this a problem?

- A "lazy" classifier that always predicts "healthy" gets 99% accuracy!
- But it completely misses all disease cases
- Accuracy alone becomes misleading

When Classes Are Imbalanced

Many datasets have unequal class distributions!

Example: Example: Medical Screening

Out of 1000 patients tested:

- **990** patients are healthy (negative class)
- **10** patients have the disease (positive class)

Key Points

Why is this a problem?

- A “lazy” classifier that always predicts “healthy” gets 99% accuracy!
- But it completely misses all disease cases
- Accuracy alone becomes misleading

Cancer Detection: Comparing Two Classifiers

Cancer Detection: Comparing Two Classifiers

Dummy Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	0	0
	Neg	10	990

Cancer Detection: Comparing Two Classifiers

Dummy Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	0	0
	Neg	10	990

- Precision: N/A

Cancer Detection: Comparing Two Classifiers

Dummy Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	0	0
	Neg	10	990

- Precision: N/A
- Recall: 0%

Cancer Detection: Comparing Two Classifiers

Dummy Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	0	0
	Neg	10	990

- Precision: N/A
- Recall: 0%
- Accuracy: 99%

Cancer Detection: Comparing Two Classifiers

Dummy Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	0	0
	Neg	10	990

- Precision: N/A
- Recall: 0%
- Accuracy: 99%

Smart Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	8	40
	Neg	2	950

Cancer Detection: Comparing Two Classifiers

Dummy Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	0	0
	Neg	10	990

- Precision: N/A
- Recall: 0%
- Accuracy: 99%

Smart Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	8	40
	Neg	2	950

- Precision: $\frac{8}{48} \approx 16.7\%$

Cancer Detection: Comparing Two Classifiers

Dummy Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	0	0
	Neg	10	990

- Precision: N/A
- Recall: 0%
- Accuracy: 99%

Smart Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	8	40
	Neg	2	950

- Precision: $\frac{8}{48} \approx 16.7\%$
- Recall: $\frac{8}{10} = 80\%$

Cancer Detection: Comparing Two Classifiers

Dummy Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	0	0
	Neg	10	990

- Precision: N/A
- Recall: 0%
- Accuracy: 99%

Smart Classifier

Confusion Matrix		Ground Truth	
		Pos	**Neg**
Pred	**Pos**	8	40
	Neg	2	950

- Precision: $\frac{8}{48} \approx 16.7\%$
- Recall: $\frac{8}{10} = 80\%$
- Accuracy: $\frac{8+950}{1000} = 95.8\%$

Precision vs Recall: The Trade-off

You often can't have both high precision and high recall —
improving one may reduce the other.

Precision vs Recall: The Trade-off

You often can't have both high precision and high recall — improving one may reduce the other.

Definition: High Precision

Precision vs Recall: The Trade-off

You often can't have both high precision and high recall — improving one may reduce the other.

Definition: High Precision

- Predicts positive only when confident

Precision vs Recall: The Trade-off

You often can't have both high precision and high recall — improving one may reduce the other.

Definition: High Precision

- Predicts positive only when confident
- Low false alarms ($\downarrow$ FP)

Precision vs Recall: The Trade-off

You often can't have both high precision and high recall — improving one may reduce the other.

Definition: High Precision

- Predicts positive only when confident
- Low false alarms ($\downarrow$ FP)
- May miss positives ($\uparrow$ FN)

Precision vs Recall: The Trade-off

You often can't have both high precision and high recall — improving one may reduce the other.

Definition: High Precision

- Predicts positive only when confident
- Low false alarms (↓ FP)
- May miss positives (↑ FN)
- Useful when FP is costly

Precision vs Recall: The Trade-off

You often can't have both high precision and high recall — improving one may reduce the other.

Definition: High Precision

- Predicts positive only when confident
- Low false alarms (↓ FP)
- May miss positives (↑ FN)
- Useful when FP is costly

Definition: High Recall

Precision vs Recall: The Trade-off

You often can't have both high precision and high recall — improving one may reduce the other.

Definition: High Precision

- Predicts positive only when confident
- Low false alarms ($\downarrow$ FP)
- May miss positives ($\uparrow$ FN)
- Useful when FP is costly

Definition: High Recall

- Captures most positives

Precision vs Recall: The Trade-off

You often can't have both high precision and high recall — improving one may reduce the other.

Definition: High Precision

- Predicts positive only when confident
- Low false alarms ($\downarrow$ FP)
- May miss positives ($\uparrow$ FN)
- Useful when FP is costly

Definition: High Recall

- Captures most positives
- Few misses ($\downarrow$ FN)

Precision vs Recall: The Trade-off

You often can't have both high precision and high recall — improving one may reduce the other.

Definition: High Precision

- Predicts positive only when confident
- Low false alarms ($\downarrow$ FP)
- May miss positives ($\uparrow$ FN)
- Useful when FP is costly

Definition: High Recall

- Captures most positives
- Few misses ($\downarrow$ FN)
- More false alarms ($\uparrow$ FP)

Precision vs Recall: The Trade-off

You often can't have both high precision and high recall — improving one may reduce the other.

Definition: High Precision

- Predicts positive only when confident
- Low false alarms (↓ FP)
- May miss positives (↑ FN)
- Useful when FP is costly

Definition: High Recall

- Captures most positives
- Few misses (↓ FN)
- More false alarms (↑ FP)
- Useful when FN is costly

Accuracy Metric: F1-Score

Confusion Matrix		Ground Truth	
		Positive	**Negative**
Predicted	**Positive**	TP	FP
	Negative	FN	TN

Example: F1-Score: Balancing Precision and Recall

$$\mathrm{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Accuracy Metric: Matthews Correlation Coefficient (MCC)

Confusion Matrix		Ground Truth	
		Positive	**Negative**
Predicted	**Positive**	TP	FP
	Negative	FN	TN

Example: MCC: Balanced Performance Measure

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

MCC Comparison: Dummy vs Smart Classifier

MCC Comparison: Dummy vs Smart Classifier

Dummy Classifier

Confusion Matrix		Ground Truth	
		Pos	Neg
Pred	Pos	0	0
	Neg	10	990

$\mathrm{MCC} = 0$ (denominator undefined; treat as 0)

MCC Comparison: Dummy vs Smart Classifier

Dummy Classifier

Confusion Matrix		Ground Truth	
		Pos	Neg
Pred	Pos	0	0
	Neg	10	990

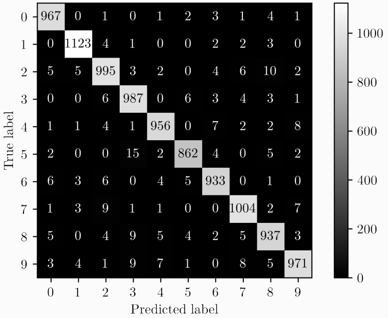
Smart Classifier

Confusion Matrix		Ground Truth	
		Pos	Neg
Pred	Pos	8	40
	Neg	2	950

$\text{MCC} = 0 \quad \text{(denominator undefined; treat as 0)}$ $\text{MCC} = \dfrac{7600}{\sqrt{(48)(10)(990)(952)}} \approx \mathbf{0.26}$

Confusion Matrix for multi-class classification

Notebook: confusion-mnist.html



Metrics for Regression: MSE & MAE

$$\text{Prediction } (\hat{y}) \quad \begin{pmatrix} 10 \\ 20 \\ 30 \\ 40 \\ 50 \end{pmatrix} \qquad \text{Ground Truth } (\mathbf{y}) \quad \begin{pmatrix} 20 \\ 30 \\ 40 \\ 50 \\ 60 \end{pmatrix}$$

$$\text{Mean Squared Error (MSE)} = \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}$$

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\text{MSE}}$$

Accuracy Metrics: MAE & ME

Prediction ($\hat{y}$)	Ground Truth
$\begin{pmatrix} 10 \\ 20 \\ 30 \\ 40 \\ 50 \end{pmatrix}$	$\begin{pmatrix} 20 \\ 30 \\ 40 \\ 50 \\ 60 \end{pmatrix}$

$$\text{Mean Absolute Error (MAE)} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}$$

$$\text{Mean Error} = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)}{n}$$

Accuracy Metrics: MAE & ME

Prediction ($\hat{y}$)	Ground Truth
$\begin{pmatrix} 10 \\ 20 \\ 30 \\ 40 \\ 50 \end{pmatrix}$	$\begin{pmatrix} 20 \\ 30 \\ 40 \\ 50 \\ 60 \end{pmatrix}$

$$\text{Mean Absolute Error (MAE)} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}$$

$$\text{Mean Error} = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)}{n}$$

Is there any downside with using mean error?

Accuracy Metrics: MAE & ME

$$\text{Prediction } (\hat{y}) \quad \begin{pmatrix} 10 \\ 20 \\ 30 \\ 40 \\ 50 \end{pmatrix} \qquad \text{Ground Truth} \quad \begin{pmatrix} 20 \\ 30 \\ 40 \\ 50 \\ 60 \end{pmatrix}$$

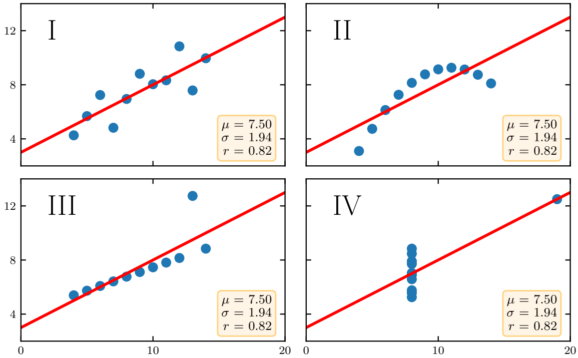
$$\text{Mean Absolute Error (MAE)} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}$$

$$\text{Mean Error} = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)}{n}$$

Is there any downside with using mean error?
Errors can get cancelled out

The Importance of Plotting

Notebook: anscombe.html



Anscombe's Quartet

Dummy Baselines

Notebook: dummy-baselines.html

The Importance of Plotting

Property	Value	Across datasets
mean(X)	9	exact
mean(Y)	7.5	up to 3 decimal places
Linear regression line	$y = 3.00 + 0.500x$	up to 2 decimal places

Pop Quiz #3

Quick Quiz 3

For imbalanced datasets, which metrics should you prioritize over accuracy?

- a) Only precision

Pop Quiz #3

Quick Quiz 3

For imbalanced datasets, which metrics should you prioritize over accuracy?

- a) Only precision
- b) Only recall

Pop Quiz #3

Quick Quiz 3

For imbalanced datasets, which metrics should you prioritize over accuracy?

- a) Only precision
- b) Only recall
- c) Precision, recall, and F1-score

Pop Quiz #3

Quick Quiz 3

For imbalanced datasets, which metrics should you prioritize over accuracy?

- a) Only precision
- b) Only recall
- c) Precision, recall, and F1-score
- d) Only confusion matrix

Pop Quiz #3 - Answer

Answer: c) Precision, recall, and F1-score give a more complete picture!

Key Takeaways

Key Points

- **ML vs Traditional Programming:** ML learns rules from data, traditional programming uses predefined rules

Key Takeaways

Key Points

- **ML vs Traditional Programming:** ML learns rules from data, traditional programming uses predefined rules
- **Features matter:** Choose meaningful features, avoid arbitrary identifiers

Key Takeaways

Key Points

- **ML vs Traditional Programming:** ML learns rules from data, traditional programming uses predefined rules
- **Features matter:** Choose meaningful features, avoid arbitrary identifiers
- **Classification vs Regression:** Discrete outputs vs continuous outputs

Key Takeaways

Key Points

- **ML vs Traditional Programming:** ML learns rules from data, traditional programming uses predefined rules
- **Features matter:** Choose meaningful features, avoid arbitrary identifiers
- **Classification vs Regression:** Discrete outputs vs continuous outputs
- **Accuracy isn't everything:** For imbalanced data, use precision, recall, F1-score

Key Takeaways

Key Points

- **ML vs Traditional Programming:** ML learns rules from data, traditional programming uses predefined rules
- **Features matter:** Choose meaningful features, avoid arbitrary identifiers
- **Classification vs Regression:** Discrete outputs vs continuous outputs
- **Accuracy isn't everything:** For imbalanced data, use precision, recall, F1-score
- **Visualization is crucial:** Always plot your data (Anscombe's Quartet lesson)

Key Takeaways

Key Points

- **ML vs Traditional Programming:** ML learns rules from data, traditional programming uses predefined rules
- **Features matter:** Choose meaningful features, avoid arbitrary identifiers
- **Classification vs Regression:** Discrete outputs vs continuous outputs
- **Accuracy isn't everything:** For imbalanced data, use precision, recall, F1-score
- **Visualization is crucial:** Always plot your data (Anscombe's Quartet lesson)
- **Use baselines:** Simple baseline models help validate your approach